

English Abstract of  
Japanese Unexamined Patent Application Publication No.63-  
113659

5

Publication date: May 18, 1988

Filing number: 61-259375

Filing date: October 30, 1986

Applicant: Hitachi Corporation, Hitachi Maxell Corporation,  
10 and Hitachi Ultra LSI engineering Corporation

Inventor: Kenji Kaneko et al.

Title: "Elements for Multi-Processor"

Abstract

15

As shown in Fig. 1, a host CPU 401 is connected, via a  
bus 402, to a plurality of processing elements (PEs) 411 -  
419. The host CPU 401 assigns two-dimensional PE-numbers to  
all the PEs. The two-dimensional PE-numbers include (1,1)  
20 to (m,n).

In order to communicate with the PEs, the upper-rank  
processor 401 uses information, which includes a combination  
of a two-dimensional PE-number and a command, from the  
upper-rank processor. Fig. 4(a) shows one example of the  
25 information format 501 that includes a two-dimensional PE-

number (502, 503) and a command 504, wherein the number 502 indicates the x-directional position and the number 503 indicates the y-directional position.

For example, as shown in Fig. 5(a), when the upper-rank processor 401 sends information that includes the PE-number (0,0) and command C, all the PEs consider that the received information is for themselves, and respond to the upper-rank processor 401. Accordingly, the upper-rank processor 401 can control all the PEs to execute the same processing operations as shown in Fig. 3(a) by transmitting the instruction only once.

As shown in Fig. 5(b), when the upper-rank processor 401 sends information that includes the PE-number (x,0) and command C, all the PEs, that are located in a column of the corresponding x-position, consider that the received information is for themselves, and respond to the upper-rank processor 401. Accordingly, the upper-rank processor 401 can control all the PEs on a desired column to execute the processing operations as shown in Fig. 3(b).

By sending information that includes some PE-number (x,y) and command C, it is possible to let only the desired PE (x,y) to respond to the upper-rank processor 401.

Each PE has components shown in Fig. 6. Alternatively, each PE may perform processing operation equivalent to the component structure of Fig. 6 by executing a software such

as a micro-program. As shown in Fig. 6, each PE has: a reception register 501, reference registers 701 and 702, comparators 704 and 707, zero detectors 703 and 706, an input portion 710, an AND gate 709, and OR gates 705 and 708.

5 The reception register 501 is for receiving the PE-number (502, 503) from the upper rank processor 601. The reference registers 701 and 702 previously store therein the PE-numbers assigned to the PE. With this structure, the OR gate 705 outputs a matching signal when the number 502 is  
10 zero or when the number 502 coincides with the number 701. The OR gate 708 outputs a matching signal when the number 503 is zero or when the number 503 coincides with the number 702. By determining a logical multiplication of the matching signals at the AND gate 709, the received two-  
15 dimensional PE-number is analyzed to determine whether or not the received command 504 should be inputted via the input portion 710.

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭63-113659

⑬ Int.Cl.<sup>4</sup>

G 06 F 15/16

識別記号

3 7 0  
4 0 0

庁内整理番号

T-2116-5B  
N-2116-5B

⑭ 公開 昭和63年(1988)5月18日

審査請求 未請求 発明の数 1 (全14頁)

⑮ 発明の名称 マルチプロセッサ用エレメント

⑯ 特 願 昭61-259375

⑰ 出 願 昭61(1986)10月30日

⑱ 発 明 者 金 子 憲 二 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑲ 発 明 者 上 田 博 唯 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑳ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉑ 出 願 人 日立マクセル株式会社 大阪府茨木市丑寅1丁目1番88号

㉒ 出 願 人 日立超エル・エス・アイ・エンジニアリング株式会社  
東京都小平市上水本町1448番地

㉓ 代 理 人 弁理士 磯村 雅俊  
最終頁に続く

明 細 書

1. 発明の名称

マルチプロセッサ用エレメント

2. 特許請求の範囲

1. 上位プロセッサが複数個の下位プロセッサ・エレメントを制御するマルチプロセッサシステムにおいて、該上位プロセッサから予め設定された固有で多次元のプロセッサ番号を格納するレジスタと、該上位プロセッサからプロセッサ番号と対になったコマンドが送られたとき、該プロセッサ番号と上記レジスタに格納されている番号とを次元ごとに比較する手段と、該比較手段による比較結果が一致するか、該プロセッサ番号が各次元ごとに予め定められた特殊な数値であるときのみ、該コマンドを受け入れて応答する手段を有することを特徴とするマルチプロセッサ用エレメント。

2. 上記プロセッサ番号は、多次元のベクトルとして認識され、該ベクトルを照合することにより自分宛のコマンドであるかを判定するこ

とを特徴とする特許請求の範囲第1項記載のマルチプロセッサ用エレメント。

3. 上記プロセッサ番号は、ビットマップ方式として認識され、該ビットマップを照合することにより自分宛のコマンドであるかを判定することを特徴とする特許請求の範囲第1項記載のマルチプロセッサ用エレメント。

4. 上記プロセッサ番号は、次元ごとの切れ目のビット位置を複数に切り換えることを特徴とする特許請求の範囲第1項、第2項または第3項記載のマルチプロセッサ用エレメント。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、マルチプロセッサ用エレメントに関し、特に超高速のデータ処理を行うための複数個の信号処理用プロセッサ・エレメントに関するものである。

〔従来の技術〕

マルチプロセッサ構成でデータ処理を行う場合に、配列された複数のプロセッサに対してそれぞ

れ異なる仕事を行わせたり、上位のプロセッサと各プロセッサとの間でコミュニケーションを行ったりするために、各プロセッサにはプロセッサ番号の割付けが必要である。

従来、プロセッサに対する番号の割り付け方法としては、例えば“日経エレクトロニクス1984.4/9、p206に記載されているような形態がある。この例では、各プロセッサの外部に各々のプロセッサ番号設定用のバッファを設けておき、システムの初期設定時に、1次元の番号を各プロセッサの内部レジスタに記憶させる方法が用いられている。すなわち、上述のマルチプロセッサは、複数個のイメージパイプラインプロセッサをリング状に接続している。各プロセッサのモジュールナンバーレジスタは4ビットからなり、16通りのモジュール番号がある。プロセッサ間の接続は、前段のプロセッサの出力系のバスと次段のプロセッサの入力系のバスを単に結線して、モジュール番号設定用の4ビットバッファを付加するだけでよい。このバッファの値は、初期設定時にプロセッサ内

にセットされ、各プロセッサの固有のモジュール番号となる。

第2図は、従来のマルチプロセッサを構成するプロセッサエレメントの配列図である。

第2図では、上位プロセッサ101のもとに、 $m \times n$ 個のデータ処理部111～119がシステムバス102を介して並列に接続されている例を示している。ここでは、2次元画像のデータ処理を行う場合を考える。2次元画像の処理では、 $M \times N$ 画素について、フィルタ演算等が行われるので、そのデータ演算量は膨大なものになる。その一例として、 $2000 \times 2000$ 画素について、 $3 \times 3$ 画素の近傍演算によるフィルタ処理を行った場合には、 $2000 \times 2000 \times 3 \times 3 = 36000000$ 回ものデータ演算が必要となる。このような演算を行う場合には、第2図に示したようなマルチプロセッサ構成で、各データ処理部に $M \times N$ 画素の領域を分割したデータを与え、各々のデータ処理部で並列に演算を行うことにより、データ処理時間がデータ処理部の数分の1に短縮

される。このとき、上位プロセッサから、どのデータ処理部にどの領域の処理を行わせるかを知らせるために、プロセッサエレメント111～119に番号付け(プロセッサエレメントのPE番号)をしておく必要がある。従来の方法では、第2図のPE内の( )で示すように、1～ $m \times n$ の連続する整数の番号を付与している。

この場合、データ処理の内容が、第3図(a)に示したように、 $M \times N$ 画素領域301の全面(斜線部分)にわたって一様なときには、PE番号はユニークであればよいので、1次元的にシーケンシャルにPE番号を付ければよい。このときには、 $M \times N$ の全面画素に対して、その一部分領域の画素302ごとに処理すべきPEを割当てて、しかし、画像処理では、第3図(b)(c)(d)に示すように、 $M \times N$ 画素領域の一部分302、303、304(斜線部分)だけの処理を行う必要が生じることが度々ある。この場合には、1次元の番号付けでは、上位プロセッサが各データ領域を分担している各々のPEに対して何回も通信を行うことにより、

処理の指示を行う必要がある。このため、上位プロセッサの処理負担が増加することになる。また、システムバスを介して行われる上位プロセッサとPE間の通信量が増加するので、上位プロセッサのオペレーティングシステムの負担が増加し、かつデータ処理のスループットは低下し、その他の処理速度も低下する等の好ましくない状態となる。

例えば、第3図(b)に示すように、 $y$ 方向の部分画像領域に対して同一処理を行う場合を考えると、従来例では、その処理の手順は次のようになっている。

(イ) 上位プロセッサが処理すべき領域を分担している複数のPEに対して、処理に必要なパラメータやプログラム等の情報を転送する必要がある。その場合、PEの番号が1次元で表現されているため、上位プロセッサからの情報転送を行うという指示を、PE数分の回数だけ行う必要がある。このため、その指示の転送にかなりの時間を要することになる。また、上位プロセッサは、常にこの情報転送を監視している必要があるため、上記指

示の転送の期間中に、他の仕事をする事ができなくなる。

(ロ) 上記情報転送の後に、上位プロセッサは、各PEに対して処理の実行を指示するが、このときにも、PEの数だけ指示を発行する必要があるので、無駄な時間が費される。

〔発明が解決しようとする問題点〕

従来方法では、(a)プロセッサ番号が1次元の指定のために、画像処理を行う時のフレキシビリティに欠けること、(b)各プロセッサと上位のプロセッサとの間のコミュニケーションの多様な形態に関しての配慮が足りないこと、等の問題があった。

本発明の目的は、これらの問題を解決し、ホストプロセッサと複数のプロセッサ間の同時通信が可能で、かつ超高速のデータ処理が可能なマルチプロセッサ用エレメントを提供することにある。

〔問題点を解決するための手段〕

上記目的を達成するため、本発明のマルチプロセッサ用エレメントは、上位プロセッサから予め

設定された固有で多次元のプロセッサ番号を格納するレジスタと、該上位プロセッサからプロセッサ番号と対になったコマンドが送られたとき、該プロセッサ番号と上記レジスタに格納されている番号とを次元ごとに比較する手段と、該比較手段による比較結果が一致するか、該プロセッサ番号が各次元ごとに予め定められた特殊な数値であるときのみ、該コマンドを受け入れて応答する手段を有することに特徴がある。

〔作用〕

本発明においては、複数のプロセッサに割り付けるプロセッサ番号を、次のような方法とする。

(a) プロセッサ番号を多次元の番号とする。これにより、画像処理等における2次元、3次元あるいは画像領域の任意の領域のデータ処理を各プロセッサに容易に分担させることができる。

(b) プロセッサ番号の特殊な番号(例えば、ゼロ等)を上位プロセッサとのグローバル通信のための番号として設ける。これにより、上位プロセッサと複数のプロセッサが同時に通信できる。

(c) プロセッサ番号のフォーマットを複数にして、上位プロセッサからの指令でフォーマットの変更ができるようにする。これにより、画像処理等において、複数の各プロセッサに処理すべき画像領域のデータを1次元的に割り当てたり、2次元的に割り当てたりすることができ、処理形態のフレキシビリティが増大する。

〔実施例〕

以下、本発明の実施例を、図面により詳細に説明する。

第1図は、本発明の一実施例を示すマルチプロセッサ用プロセッサエレメントの配置図である。

第1図においては、ホストCPU401は、バス402を介して接続されている複数のPEに対して、2次元のPE番号、つまり(1, 1)から(m, n)までの2桁ずつの番号を与えている。

上位プロセッサ401とPE411~419の間での通信を行うには、この2次元のPE番号と上位プロセッサからのコマンドを対にした情報を用いて行う。

第4図は、本発明によるPE番号のフォーマット例を示す図である。第4図(a)では、このPE番号502, 503とコマンド504を対にした情報のフォーマット501を示している。PE番号502, 503は、各々行(x方向)、列(y方向)の番号を意味している。

この情報は、システムバスのビット長に応じて決められ、例えば16ビット長の場合、コマンドに4ビットを用いて、残り12ビットをPE番号502, 503に割り付ける。このとき、第4図(b)に示すように、12ビットを6ビットごとにx, yの番号に割り当ててもよく、また第4図(c)に示すように、異なるビット数分だけ割り当ててもよい(3ビットと9ビット)。

本実施例においては、x, yへの番号の割り当ては、数種類準備し、上位プロセッサからのモード設定の変更により、その時々により番号割り当て方法を変更することができるようにしている。

このモード設定は、各PEにモード設定用レジスタを設けておき、これに上位プロセッサから、

モード設定データを書き込むようにしている。このモード設定データにより、上位プロセッサから送られてくるPE番号のビット配分を認識する。このようにすると、行、列の方向の画素が同程度の場合や、片方向だけ大きい場合（極端な場合には、1次元画素データ）にも、簡単にプロセッサの割り付け変更ができる。

本実施例では、上述のように、多次元に番号割り付けを行った上で、さらにPE番号のうち特殊な数を上位プロセッサとPE間のグローバル通信に用いる。ここでは、一例として、ゼロをグローバル通信用の数とする。このようにすることにより、第3図(a)(b)(c)に示したような形態のデータ処理が容易になる。

第5図は、本発明におけるホストCPUと同時に通信可能なPEの例を示す図である。

第5図(a)に示すように、上位プロセッサ601からPE番号(0, 0)とコマンドCが対になって送られてきたとき、全てのPEは同時に自分に送られてきたコマンドであると解釈して、上

位プロセッサに応答する。このため、第3図(a)に示すような全面同一処理のための上位プロセッサからの指示が、1回で済むことになる。

また、第5図(b)に示すように、PE番号(x, 0)とコマンドCが対になって送られてきたときには、あるx行のプロセッサの全てが同時に応答し、第3図(b)に示す行方向のデータ処理が容易に行われる。同じようにして、y列方向の処理も容易に行われることは勿論である。また、PE番号(x, y)を送ることにより、特定のPEだけが応答することも勿論容易である。

このように、本実施例においては、上位プロセッサから複数のPEに対する指示は、指示内容ごとに1回の指示だけで済むので、このための通信時間が大幅に短縮される。

第6図は、本発明のPE内の2次元PE番号を判定する回路の一実施例を示す構成図である。

第5図に示すような上位プロセッサとPE間の通信を実現するためには、第6図で示すような構成要素を各PE内に備えるか、または第6図に示

す構成での処理手順をマイクロプログラム等によるソフトウェアにより行うことによって可能となる。

第6図において、501は受信レジスタ、701, 702は比較のための基準レジスタ、704, 707は比較器、703, 706はゼロ検出回路、710は入力部、709はANDゲート、705, 708はORゲートである。

上位プロセッサから送られてきた情報502, 503, 504（順に、x方向、y方向、コマンド）は、PEにおいて、自分宛の情報か否かが判定される。すなわち、PEは、送られてきたx方向番号502、y方向番号503がゼロか、あるいは送られてきたx方向番号502、y方向番号503が、それぞれ基準レジスタに格納され、自分に割り当てられている2次元のPE番号701, 702と一致するか否かを比較器704, 707で比較する。ゼロ検出器703と比較器704での判定結果は、論理的にORゲート705で論理和がとられる。つまり、502の番号がゼロであ

るか、あるいは502と701の番号が一致した場合に、一致信号が出力される。同じようにして、503の番号がゼロであるか、あるいは503と702の番号が一致した場合に、一致信号が出力される。これら両方の一致信号をANDゲート709で論理積をとることにより、x行、y列方向での2次元のPE番号の判定が行える。そして、ANDゲート709での一致信号を用いて、上位プロセッサからのコマンド504を入力部710から入力するか否かを決定する。

第7図は、本発明の他の実施例を示すもので、3次元データ処理を行う場合のPE番号の判定法を示す図である。3次元データの場合には、第1図におけるマルチプロセッサの構成を3次元とし、各PEに3次元のPE番号(x, y, z)を付与する。この番号のうち、ゼロは2次元の場合と同じように、グローバル通信用の番号とする。

第7図における3次元番号の判定は、上位プロセッサから送られてきたPE番号804, 805, 806を、PE内に設定されている番号801,

802, 803と比較した結果と、送られてきたPE番号804, 805, 806がそれぞれゼロであるか否かの判定結果との論理和、論理積をとって、その出力信号によりコマンド807を入力するか否かを決定する。

このように3次元のPE番号を与えた場合には、ある特定のPEだけと通信したり、ある軸(x, y, zのいずれか1つ)の番号をゼロにすることにより、1軸方向の全PEと同時に通信したり、2軸(x, y, zのいずれか2つ)の番号をゼロにすることにより、ある平面方向の全PEと同時に通信したり、3軸の番号をゼロにすることにより、全PEと同時に通信することが可能となる。

第8図(a)(b)(c)は、本発明のさらに他の実施例を示すもので、PE番号の割り当てにビットマップ方式を用いた場合の説明図である。

第8図の実施例では、 $m \times n$ 個のPEに対して2次元のPE番号を与える場合、コマンド504と対になり、情報の全体フォーマット501を形成する行(x方向)、列(y方向)のPE番号502、

503に各々mビット長、nビット長の領域を割り当て、ビットマップ方式のPE番号割り当てを行っている。この割り当て方法は、x行、y列の指定にビット数が余分に必要となるが、個々の行および列について、コマンド受信に対する選択、あるいは非選択を任意に決定できるので利用度が高い。第8図(a)(b)(c)では、行方向のPE番号を表すmビット長の領域502のaビット目( $1 \leq a \leq m$ )の論理値"0", "1"により、a行目の非選択、選択を決定し、列方向も同じようにPE番号を表すnビット長の領域503のbビット目( $1 \leq b \leq n$ )の論理値"0", "1"により、b行目の非選択、選択を決定している。そして、行方向および列方向が共に選択されたPEのみが上位プロセッサとの通信において選択される。

このようなプロセッサPEの制御方式を採用することにより、前述の第3図(b)(c)(d)に示したような形態のデータ処理に加えて、第8図(a)(b)(c)に示すようなブロック部分領域の組み合わせ形態のデータ処理が簡単に行える。なお、第8

図(a)は、第3図(d)と同一の形態であり、第8図(b)は分離された2つのブロック部分領域の組み合わせ形態のデータ処理であり、第8図(c)は4つに分離されたブロック部分領域の組み合わせ形態のデータ処理である。

第9図は、第8図の各PEにおけるPE番号判定回路のブロック図である。

第8図(a)(b)(c)で示したような上位プロセッサとPE間の通信を可能にするには、第9図に示す構成要素を各PE内に備えるか、または第9図で示す構成の処理手順をマイクロプログラム等によるソフトウェアにより行うことにより、実現される。

第9図において、上位プロセッサから送られてきた情報502, 503, 504(それぞれ、x方向番号、y方向番号、コマンド)は、PEにおいて、自分宛の情報か否かが判定される。すなわち、PE番号判定回路では、自分に割り当てられた2次元のPE番号1001, 1002のデコーダ出力信号1007, 1008と、外から送られ

てきたPE番号のビットパターンの比較を、ビットパターン比較ブロック1003, 1004により行う。このとき、 $m \times n$ 個のPEの中の座標(a, b)( $1 \leq a \leq m, 1 \leq b \leq n$ )で表わされるPEは、自分自身に割り当てられた2次元のPE番号1001, 1002として、a, bを持ち、PE番号のデコーダ1007, 1008の出力信号は、mビット長(1007の場合)、nビット長(1008の場合)の領域中でaビット目(1007の場合)、bビット目(1008の場合)のみが論理"1"、他の全ビットが論理値"0"であるビットパターンを有する。

ビットパターンが一致したならば、ANDゲート1005で両比較ブロック1003, 1004の出力の論理積をとって、ANDゲート1005の出力で入力部1006を制御することにより、コマンド504を自分のPEに入力する。

第10図は、第9図の比較ブロックの詳細構成図である。ビットパターン比較ブロック1003の構成は、第10図に示すように、上位プロセッ



サから送られた情報502と、デコーダ1007の出力信号とのビットパターンを比較するため、論理ANDゲート1101~1104をmビット分だけ並列に備えて、これらの出力をm入力の論理ORブロック1105に入力し、その出力をビットパターン比較ブロック1003の出力とする。デコーダ1007の出力信号は、前述のようにaビット目のみが論理値“1”、他の全ビットが論理値“0”であるため、ビットパターン比較ブロック1003の出力信号は上位プロセッサから送られた情報502のaビット目が論理値“1”の場合、論理値“1”に、論理値“0”の場合には“0”となる。

また、y方向のPE番号用のビットパターン比較ブロック1004も、デコーダ1008の出力信号と、上位プロセッサから送られた情報503に対して同じような動作を行う。

これら両方のビットパターン比較ブロック1003、1004の出力信号を論理的にANDゲート1005により論理積をとることにより、x行、y列方向での2次元のPE番号の判定ができる。

第11図以降は、本発明の第3の実施例を示すもので、ベクトル型のPE番号指定を採用した例を示す。ここでは、ベクトル型指定を用いて、多重次元への適用の実施例を述べる。

まず、k次元の場合には、ベクトル型PE番号指定は、 $P(i_1, i_2, \dots, i_k)$ により指定することができる。第1図の実施例では、一例として $k=2$ 、 $1 \leq i_1, i_2 \leq 8$ の場合を述べた。多重次元の場合、一般的には、

$$P(i_1(j_1, j_2, \dots, j_d), i_2(j_1, j_2, \dots, j_d), \dots, i_k(j_1, j_2, \dots, j_d))$$

によりk次元の各々の次元をさらにd多重次元で表現( $1 \leq j \leq d$ )できる。これを簡単のために、

$$P(i_1, j_1, j_2, \dots, j_d, i_2, j_1, j_2, \dots, j_d, \dots, i_k, j_1, j_2, \dots, j_d)$$

と記載する。第11図は、本発明のベクトル型番号指定において、 $k=2$ 、 $d=2$ とした場合の実施例を示す図である。第11図では、16個のプロセッサに対して、各PEに、

$$P(i_1, j_1, i_2, j_2)$$

上記ANDゲート1005での一致信号を用いてコマンド入力ブロック1006を制御することにより、上位プロセッサからのコマンド504を入力するか否かを決定する。

また、本実施例においては、情報502および503の全ビットが論理値“1”となるビットパターンが、全PEに対するグローバルなPE選択パターンとなる。この判定には、専用のハードウェアまたはソフトウェアを用意せず実現できる。

本実施例では、2次元のPE番号の割り当てをビットマップ方式により実現したものであるが、ビットマップ方式を用いて3次元以上のさらに多次元のPE番号の割り当ても可能である。

また、PE番号の割り当てにビットマップ方式を採用することによって、第1図の実施例では不可能であった第8図(a)(b)(c)に示すブロック部分領域や、ブロック部分領域の組み合わせ形態の多様なデータ処理が簡単に行える。この結果、画像処理における部分画像の処理等を高速に行うことが可能である。

の番号を割り付ける。

第11図の例では、実線で囲まれた4分割(1分割当りのPE数が4個)1201に対して $i_1, i_2$ の番号を割付け、さらにその1分割内の点線で囲まれた4個のPE1202に対して $j_1, j_2$ の番号を割付けている。また、数値ゼロをグローバル通信用の数値とする。この場合には、第12図(a)~(e)に示すような多様な形態でのホストCPUとの通信が可能となる。

第12図(a)では、 $P(0, 0, 1, 0)$ というPE番号がホストCPUから送られてきたときに応答する8個のプロセッサを、斜線部分1301で示している。すなわち、 $P(i_1, j_1, i_2, j_2)$ のうち、 $i_1, i_2$ が0, 1であるから、x方向は全グループのPE、y方向は第1番目のグループのPEということになる。そして、 $j_1, j_2$ が0, 0であるため、斜線部分1301の全てのPEが指定されることになる。

第12図(b)では、 $P(0, 0, 1, 2)$ のPE番号がホストCPUから送られてきたときに応答する

横1列の4個のプロセッサを、斜線部分1302で示している。

第12図(c)では、 $P(0, 1, 1, 0)$ というPE番号がホストCPUから送られてきたときに応答する4個のプロセッサを斜線部1303, 1304で示している。

第12図(d)では、 $P(1, 2, 1, 0)$ というPE番号がホストCPUから送られてきたときに応答する2個のプロセッサを斜線部分1305で示している。

第12図(e)では、 $P(1, 0, 2, 0)$ というPE番号がホストCPUから送られてきたときに応答する4個のプロセッサを斜線部分1306で示している。

第12図では、5つの例を示したが、勿論他にも1個のプロセッサだけとの通信( $i_1, j_1, i_2, j_2$ の値を全てゼロとしない)、あるいは全プロセッサとの同時通信( $i_1, j_1, i_2, j_2$ の値を全てゼロにする)も可能である。

また、第11図、第12図の実施例において、

を行い(ステップ3)、No.のときには何もせず(ステップ4)、Yes.のときのみ、コマンドを取り込む(ステップ5)、コマンドの内容の実行、つまりRUN状態に移行する(ステップ6)。

第14図は、第14図の下位プロセッサにおける入力制御部のアドレス、命令の取り込み動作の説明図である。各下位プロセッサ10, 10'は、第14図(a)に示すように、データストロープDSとアドレス信号Aリード/ライト信号R/Wが外部から入出力制御部に入力されることにより、出力レジスタ12から出力データが外部バスに出力され、また入力レジスタ13へ外部からの入力データを取り込む。第15図(b)は、(a)の入出力制御部のさらに詳細な構成を示すもので、リード/ライト信号R/Wとデータストロープ信号DSとアドレス信号AとをANDまたはNANDゲートを経由して出力および入力ゲートに供給することにより、共通外部バスを経由してデータを入力レジスタ13にセットするか、あるいは出力レジスタ12のデータを共通外部バスに出力する。

ホストCPUとの通信を行う場合に、PE番号とコマンドとを対にしたデータを用いて情報通信を行うことは、それ以前に述べた実施例の場合と同じである。

さらに、PE番号の判定についても、それ以前に述べた実施例と同じ方法で実現できることは勿論である。

このように、第11図の実施例においては、第1図の実施例では不可能であったブロック領域の複数プロセッサとの通信(第12図(a)~(e))が可能となる。

第13図は、上位プロセッサと下位プロセッサの通信インタフェースの流れを示すフローチャートである。まず、ホストCPUは、コマンドとプロセッサ番号を発行する(ステップ1)。例えば、 $(1, 1, \text{RUN})$ のコマンドを、共通バスに出力することにより、これを受け取ったx方向1番目、y方向1番目のPEのプログラムが走行する。 $(1, 1)$ のPEは、まずPE番号とコマンドを一時入力する(ステップ2)。次に、PE番号の判定

第15図(c)は、それらの信号タイムチャートである。共通外部バス上に上位プロセッサから送られてきたPE番号とコマンドの対のデータが乗せられた後データストロープDSがローレベルになる。次に、データストロープDSがローレベルからハイレベルに変化した時点で、共通バス上のデータがPE内に取込まれPE内のテンポラリレジスタにこれらのデータをセットする。このようにして、上位プロセッサは指定されたPEに対して効率よく通信を行い、処理を割り当てることが可能である。

第1図の第1の実施例と、第8図の第2の実施例と、第11図の第3の実施例とを比較すると、ホストCPUとの通信の多様性という点では、第8図、第11図、第1図の順で優れている。しかしながら、通信に必要なPE番号を指定する情報量(PE番号指定に必要なビット数)を同一とした場合に、通信可能となるPE数の大きさは、第1図、第11図、第8図の順で優れている。

また、各PEに対するPE番号の設定の方法に

ついて述べると、本実施例では、従来の方法と同じように、各PEの外部に番号設定用のハードウェアを付加することにより設定できる。しかし、この他にも、各PEから上位プロセッサに対して番号設定要求を出させた上で、上位プロセッサが物理的に端にあるPEから順に論理的なPE番号を設定していくことも可能である。後者の方法は、各PEに対するPE番号の設定変更が簡単になるという利点を有する。

#### 【発明の効果】

以上説明したように、本発明によれば、マルチプロセッサ・システムにおいて、上位プロセッサから複数の下位プロセッサPEに対して、多次元のPE番号を割り付けることにより、次のような利点がある。

(a) PE番号の特殊な数値をグローバル通信用に設定することにより、上位プロセッサと複数の下位プロセッサとの同時通信が可能である。

(b) PEの2次元配列においては、特定の1個のPE、1列分のPE、1行分のPE、全てのPE、

あるいは特定の矩形領域のPEに対して、上位プロセッサは同時通信が可能である。

(c) PEの番号モードを複数設けることにより、物理的な接続を変更せずに、PEを2次元的または1次元的な配列として扱うことが可能である。

#### 4. 図面の簡単な説明

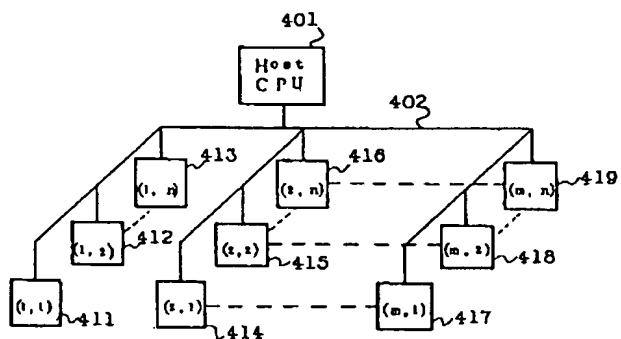
第1図は本発明の第1の実施例を示すマルチプロセッサ・システムの構成図、第2図は従来のマルチプロセッサ・システムの構成図、第3図は複数のPEの動作形態を示す図、第4図は第1図に対応するPE番号のフォーマット例を示す図、第5図はホストCPUと同時に通信可能なPEの配列を示す図、第6図は第1図のPE内で2次元PE番号を判定する回路の構成図、第7図は第1図のPE内で3次元PE番号を判定する回路の構成図、第8図は本発明の第2の実施例を示すPE番号指定方式を示す図、第9図は第8図におけるPE番号を判定する回路の構成図、第10図は第9図のビットパターン比較ブロックの詳細構成図、第11図は本発明の第3の実施例を示す多重次元

PE番号割り付け方法の図、第12図は第11図における同時通信が可能なPEの形態を示す図、第13図は上位と下位プロセッサ間の通信インタフェースの流れを示す動作フローチャート、第14図は下位プロセッサにおける入出力時の動作説明図である。

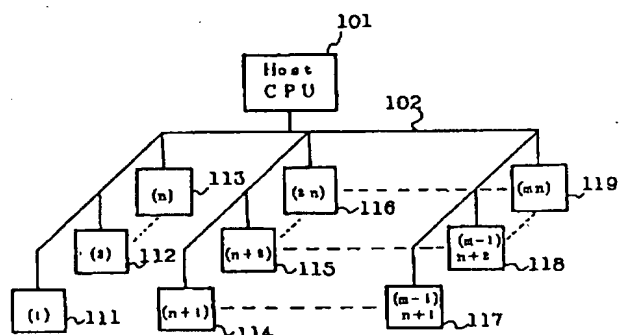
101, 401, 601: ホストCPU (上位プロセッサ)、102, 402, 602: システムバス、111~119, 411~419, 611~619: プロセッサエレメント、301: M×N全国像領域、302: 縦方向の部分領域、303: 横方向の部分領域、304: ブロック部分領域、320: 1PEのデータ処理領域、501: 上位プロセッサからの情報、502: x方向のPE番号、503: y方向のPE番号、504: コマンド、701: PEに設定されたx方向のPE番号、702: PEに設定されたy方向のPE番号、703, 706: ゼロ判定ブロック、704, 707: 番号比較ブロック、705, 708: 論理ORゲート、709: 論理ANDゲート、710

: コマンド入力ブロック、801: PEに設定されたx方向のPE番号、802: PEに設定されたy方向のPE番号、803: PEに設定されたz方向のPE番号、804~806: 上位プロセッサからのx, y, zのPE番号情報、807: 上位プロセッサからのコマンド情報、808, 811, 814: ゼロ判定ブロック、809, 812, 815: 番号比較ブロック、810, 813, 816: 論理ORゲート、817: 論理ANDゲート、901~906: ブロック部分領域(1~6)、1001, 1002: PEに設定されたx, y方向のPE番号、1003, 1004: ビットパターン比較ブロック、1005: 論理ANDゲート、1006: コマンド入力ブロック、1007, 1008: デコーダブロック、1101~1104: 論理ANDゲート、1105: 論理ORゲート、1201: 4分割されたプロセッサ群、1202: 1個のプロセッサ、1301~1306: ホストCPUに回答するプロセッサ群。

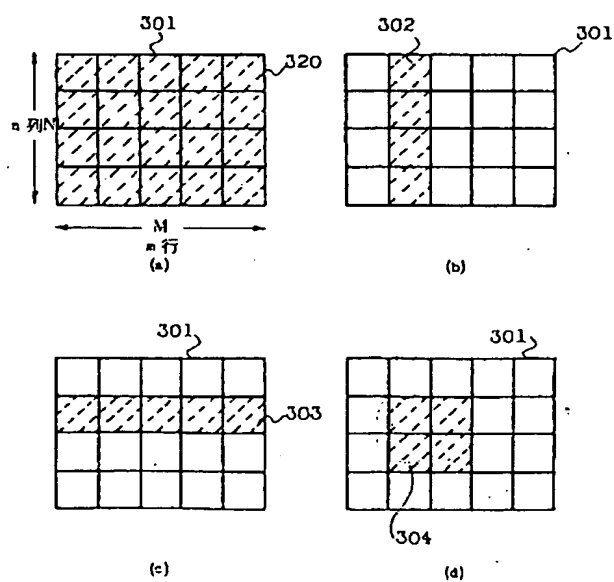
第 1 圖



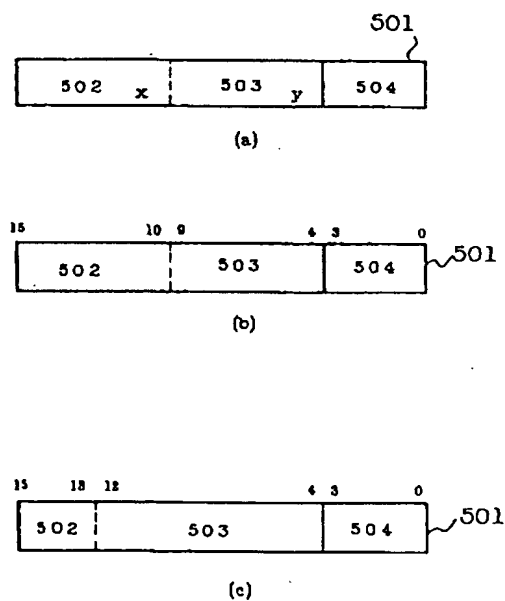
第 2 圖



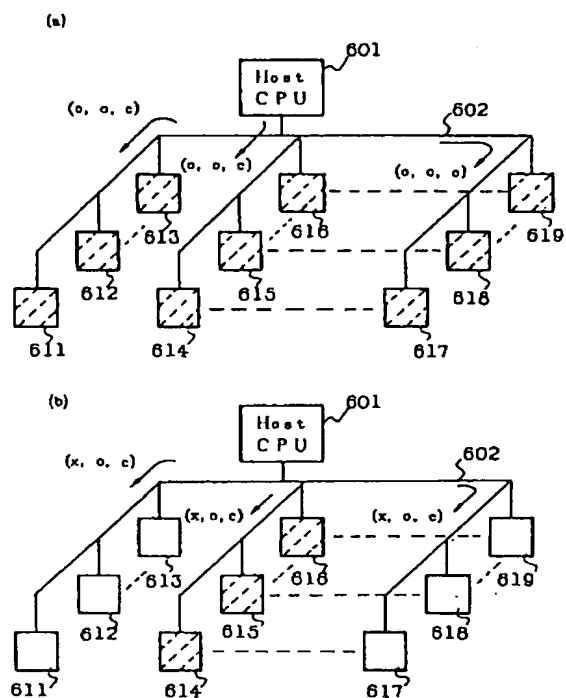
第 3 圖



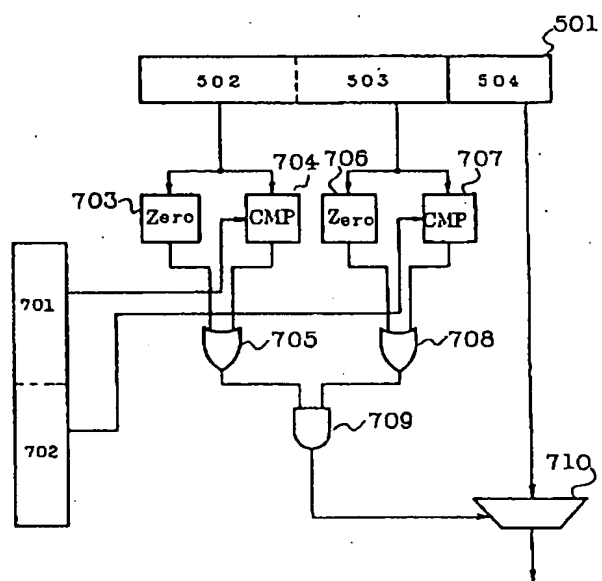
第 4 圖



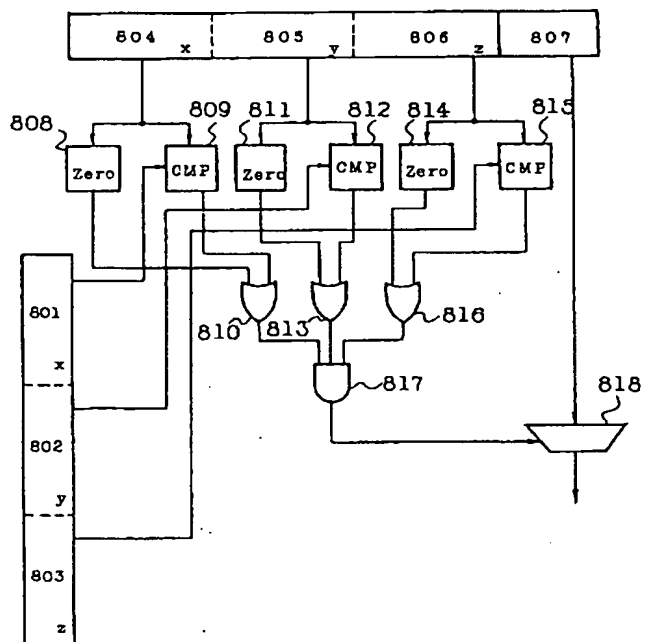
第 5 圖



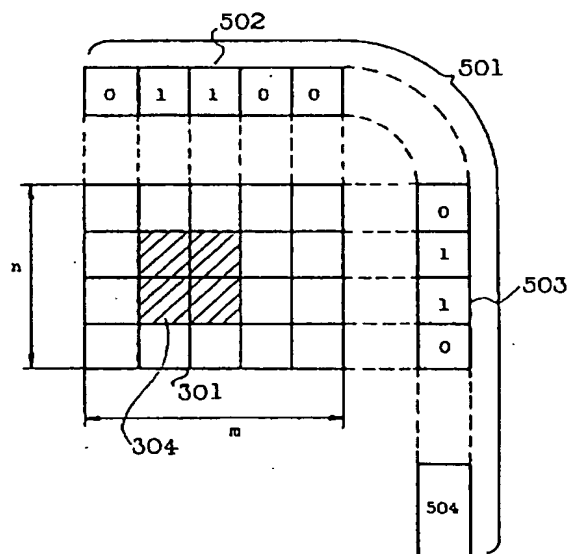
第 6 圖



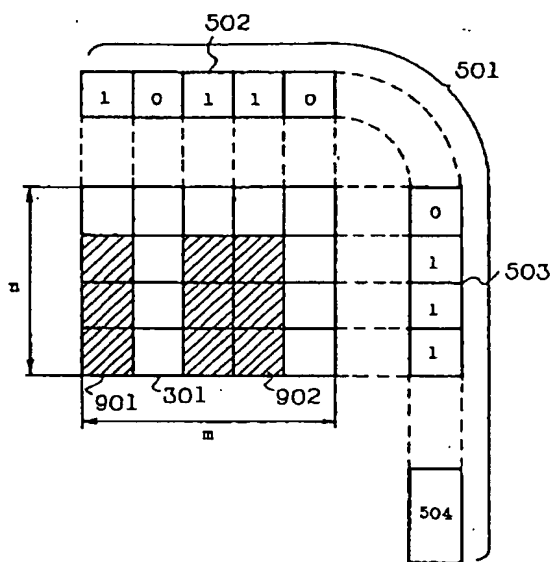
第 7 圖



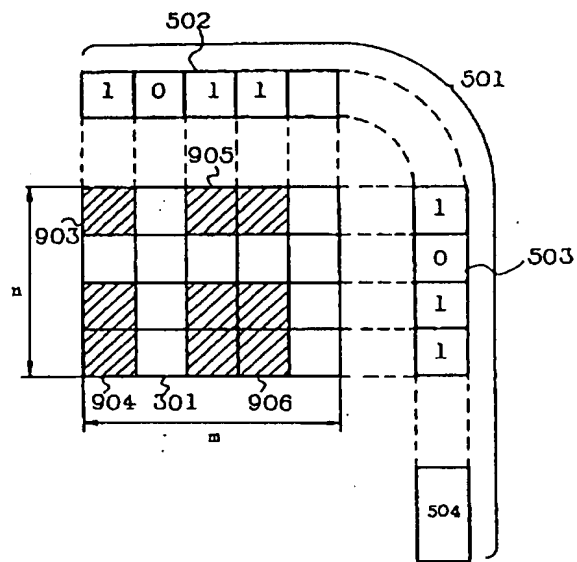
第 8 圖 (a)



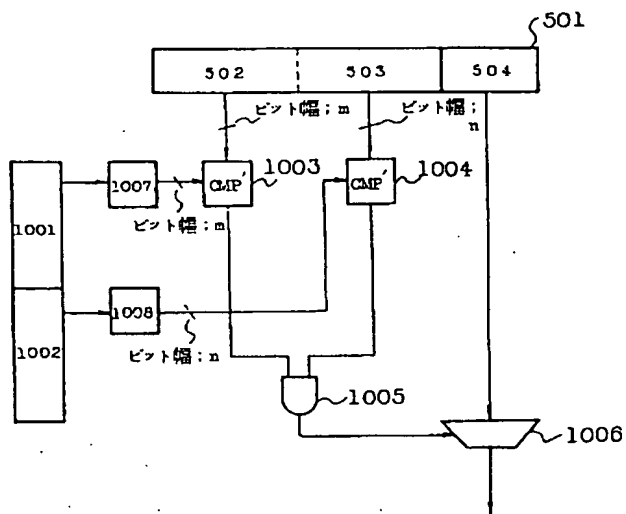
第 8 図 (b)



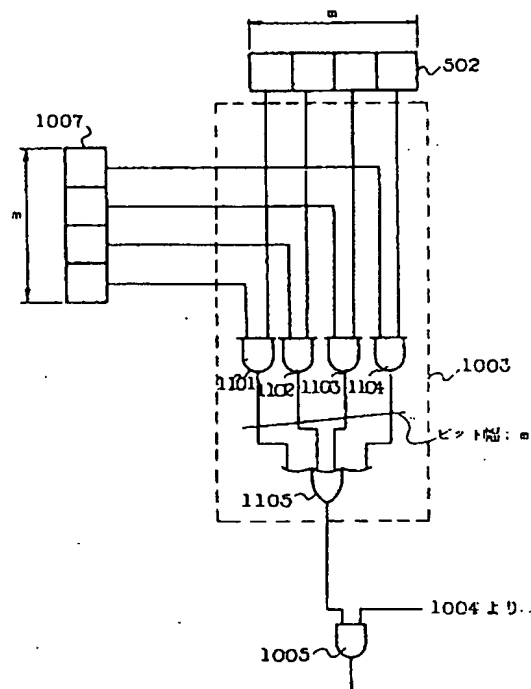
第 8 図 (c)



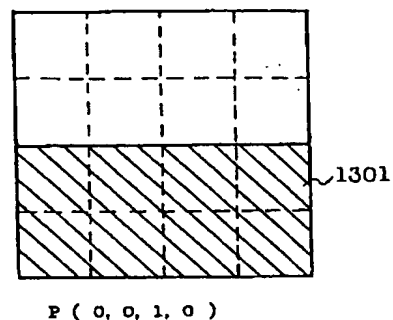
第 9 図



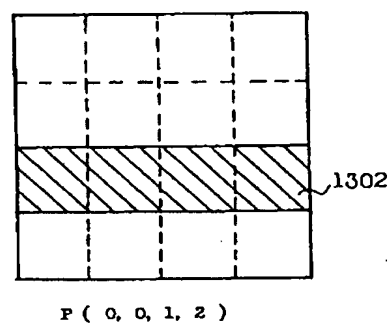
第 10 図



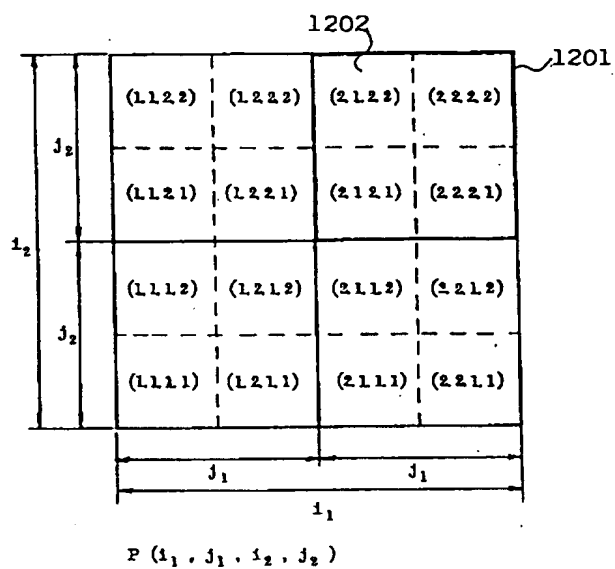
第 1 2 図 (a)



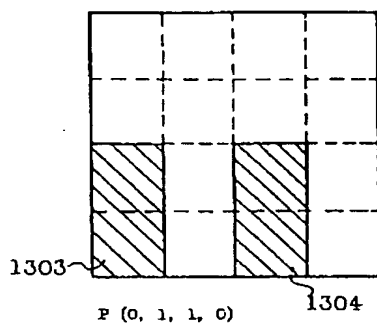
(b)



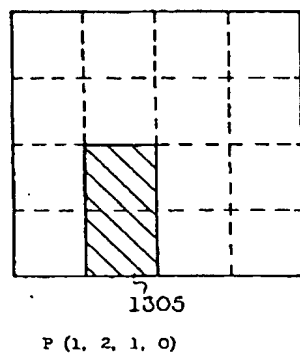
第 1 1 図



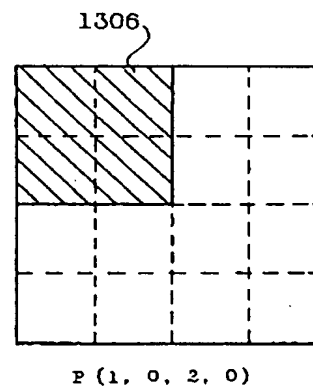
第 1 2 図 (c)



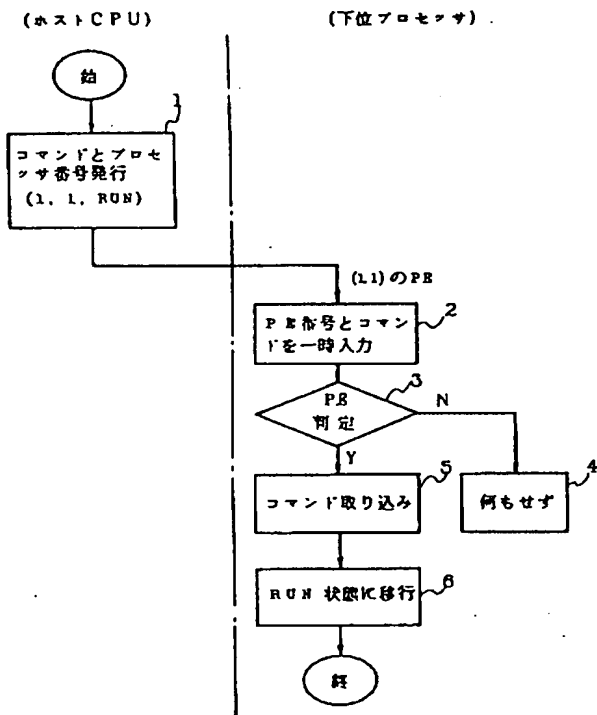
(d)



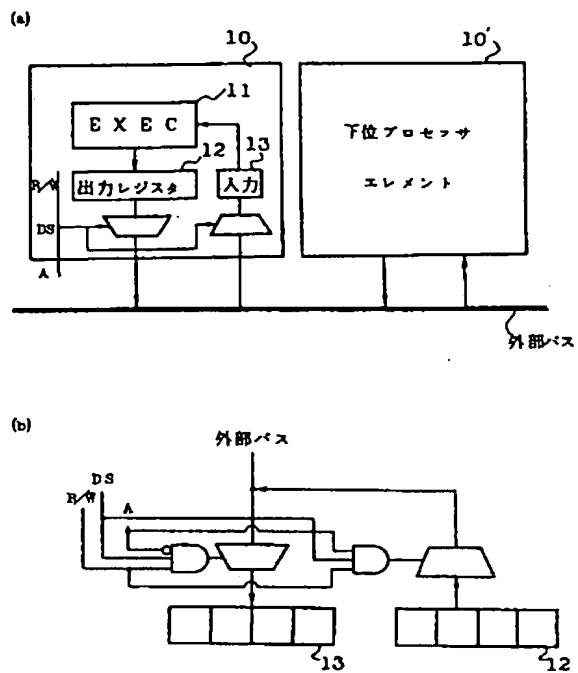
第 1 2 図 (e)



第 1 3 図

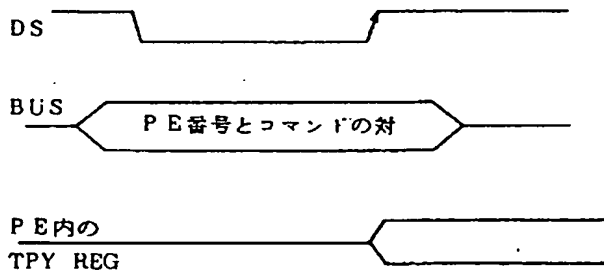


第 1 4 図



第 1 4 図

(c)





## 第1頁の続き

- ⑫発明者 中 川 哲 也 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
- ⑬発明者 木 内 淳 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
- ⑭発明者 萩 原 吉 宗 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
- ⑮発明者 高 森 洋 大阪府茨木市丑寅1丁目1番88号 日立マクセル株式会社内
- ⑯発明者 豊 増 考 乃 東京都小平市上水本町1448番地 日立超エル・エス・アイ・エンジニアリング株式会社内